

SYSTEM AND METHOD FOR SHARING RESOURCE

Patent number: JP10049390

Publication date: 1998-02-20

Inventor: MILITO RODOLFO A

Applicant: AT & T CORP

Classification:

- international: G06F9/46; H04L12/28; H04L12/56; H04M3/00;
H04Q3/00; H04Q3/545

- european:

Application number: JP19960290975 19961101

Priority number(s):

Also published as:



EP0772324 (A:

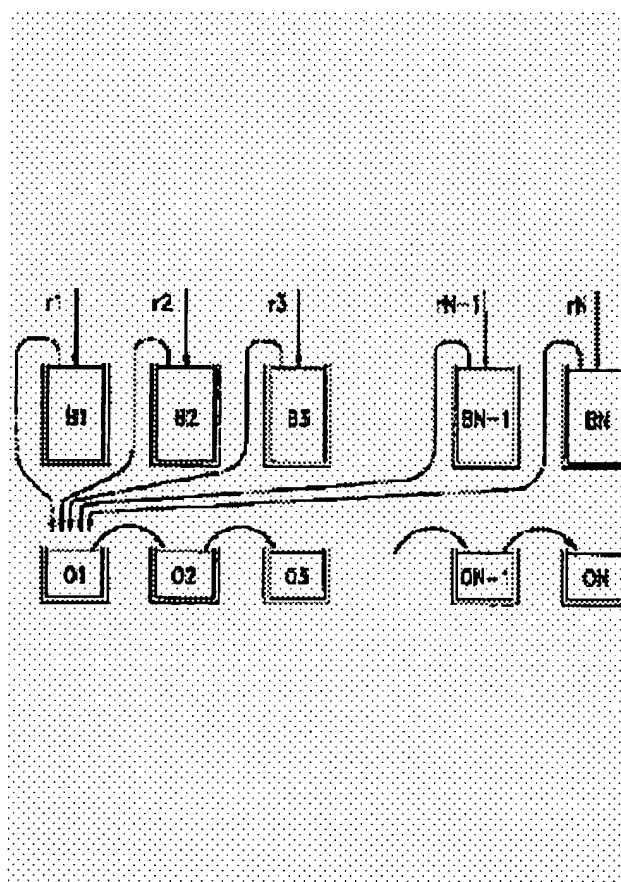
US5596576 (A:

EP0772324 (A:

Abstract of JP10049390

PROBLEM TO BE SOLVED: To assign an access to the resource that is limited and shared by plural users having access requests to the resource by permitting the accesses to the resource as long as a prescribed number of tokens are included in a related bank.

SOLUTION: The bank B_i of every class receives a token stream at a contracted rate r_i , and the overflow bank O_i is related with every class. If a bank B_i that is filled with the tokens generated to a specific class is detected, these tokens are not immediately put into the bank O_i of the relevant class. Then the extra tokens are to be put into a bank O_1 of higher priority. If the bank O_1 is filled with tokens, the extra tokens are put into an overflow bank O_2 . Then the accesses are permitted to the resource as long a prescribed number of tokens are included in a relate bank. Otherwise, the accesses are rejected to the resource.



BEST AVAILABLE COPY

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-49390

(43) 公開日 平成10年(1998) 2月20日

(51) Int. Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/46	3 4 0		G 0 6 F 9/46	3 4 0 F
H 0 4 L 12/28			H 0 4 M 3/00	D
			H 0 4 Q 3/00	
				3/545
H 0 4 M 3/00			H 0 4 L 11/00	3 1 0 D
H 0 4 Q 3/00				

審査請求 未請求 請求項の数16 O L (全 14 頁) 最終頁に続く

(21) 出願番号 特願平8-290975

(22) 出願日 平成8年(1996)11月1日

(31) 優先権主張番号 08/552899

(32) 優先日 1995年11月3日

(33) 優先権主張国 米国 (US)

(71) 出願人 390035493

エイ・ティ・アンド・ティ・コーポレーション

AT&T CORP.

アメリカ合衆国 10013-2412 ニューヨーク
ニューヨーク アヴェニュー オブ
ジ アメリカズ 32

(72) 発明者 ルドルフォ エー. ミリト

アメリカ合衆国 08854 ニュージャージー,
ビスカタウェイ, ザークル アヴェニュー 9

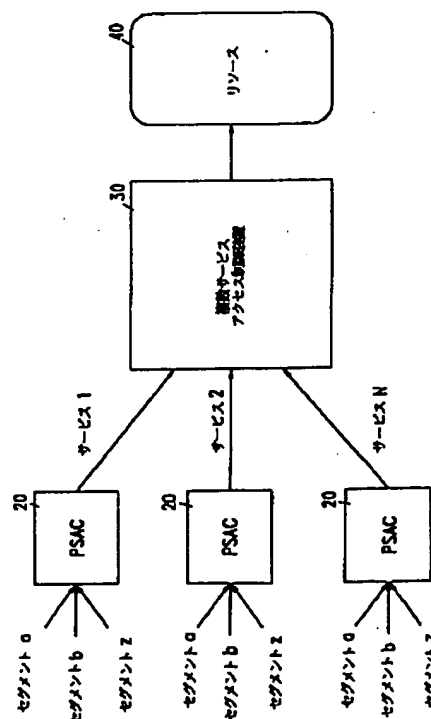
(74) 代理人 弁理士 岡部 正夫 (外1名)

(54) 【発明の名称】 リソース共用のためのシステムおよび方法

(57) 【要約】 (修正有)

【課題】 制限されたリソースを多数の使用者および／または使用者クラスの間で共用するためのシステムおよび方法に関する。

【解決手段】 各使用者に関係付けられた最小限の保証されたレートによって決定されるレートでトークンを各使用者に割り当て、一方各使用者に関係付けられたバンクに各使用者に割り当てられたトークンを蓄積して、各バンクは各使用者に関係付けられた有限の容量を持っている。その関係付けられたバンクの容量に到達した使用者に割り当てられたトークンを、そのバンクの容量が小さい他の使用者のバンクに蓄積する。使用者があらかじめ決められた数のトークンをその関係付けられたバンクに備えている場合は、リソースにアクセスを要求している使用者にそのリソースへのアクセスをさせて、さもなければリソースへのアクセスを拒絶する。



【特許請求の範囲】

【請求項1】 共用される限定されたリソースにアクセスを要求する複数の使用者の間で前記リソースにアクセスを割り当てるための方法であって、前記割り当てを、各使用者と関係付けられた最小限の保証されたレートでの前記リソースへのアクセスを各使用者が許可されるように成して、そして前記リソースが利用できる場合だけはその最小限の保証されたレートを越えているいかなる使用者によっても前記リソースが利用できるように成して、前記方法は、

各使用者に関係付けられた最小限の保証されたレートによって決定されるレートで各使用者にトークンを割り当てるステップと、

各使用者に割り当てられたトークンを各使用者に関係付けられたバンクに蓄積するステップと、ここで前記各バンクは各使用者に関係付けられた有限の容量を備えていて、

その関係付けられたバンクがその容量に到達した使用者に割り当てられたトークンを、そのバンクがその容量に到達していない別の使用者に割り当てられたバンクに蓄積するステップと、そして、

前記使用者がその関係付けられたバンクにあらかじめ決められた数のトークンを持っている場合は、前記リソースにアクセスを要求している使用者に前記リソースへアクセスすることを許可し、そうでない場合は、前記リソースへのアクセスを拒絶するステップと、から構成されることを特徴とする方法。

【請求項2】 請求項1記載の方法において、前記使用者が別々の優先権を割り当てられていて、前記別の使用者は前記割り当てられた優先権に基づいて選択されることを特徴とする方法。

【請求項3】 請求項1記載の方法においてさらに、前記リソースにアクセスを許可された使用者に関係付けられたバンクからあらかじめ決められた数のトークンを削除するステップから構成されることを特徴とする方法。

【請求項4】 請求項1記載の方法において、前記あらかじめ決められたトークンの数が1であることを特徴とする方法。

【請求項5】 請求項1記載の方法において、前記バンクの各々が、あらかじめ決められた同一の有限の容量を備えていることを特徴とする方法。

【請求項6】 請求項1記載の方法において、前記バンクの少なくとも二つが異なるあらかじめ決められた有限の容量を備えていることを特徴とする方法。

【請求項7】 共用されたリソースへのアクセスを割り当てるための方法であって、(a) 前記リソースの複数の使用者の各々に対して加算できるカウンタを確立するステップと、この前記加算できるカウンタは計数を持っていて、(b) 前記計数が各使用者に関係付けられてあらかじめ決められた最大計数値より小さい場合には、各使

用者に対して確立された各カウンタの計数を各使用者に関係付けられたレートで増加させるステップと、(c) アクセスを要求している前記使用者に対して確立されたカウンタの計数がゼロでない場合には、前記リソースにアクセスを要求している使用者にアクセスすることを許可するステップと、(d) ステップ(b)で増加させられた他のどのカウンタの計数もそのあらかじめ定められた最大計数にある場合は、そのカウンタが前記あらかじめ決められた最大計数より小さい使用者に関係付けられたカウンタの計数をその関係付けられたレートよりも高いレートで増加させるステップと、から構成されることを特徴とする方法。

【請求項8】 請求項7記載の方法において、前記使用者が異なる優先権を割り当てられていて、ステップ

(d)で選択された前記使用者が前記割り当てられた優先権に基づいて選択されることを特徴とする方法。

【請求項9】 請求項7記載の方法においてさらに、アクセスを要求している前記使用者が前記リソースへのアクセスを許可された場合は、アクセスを要求している使用者に対して確立されたカウンタの計数を削減するステップから構成されることを特徴とする方法。

【請求項10】 請求項7記載の方法において、前記各々の使用者が、その関係付けられたあらかじめ決められた同一の最大計数を備えていることを特徴とする方法。

【請求項11】 請求項7記載の方法において、前記使用者の少なくとも二つがあらかじめ決められた異なる最大計数を備えていることを特徴とする方法。

【請求項12】 N人の使用者の間での共有リソースへのアクセスを共用する方法であって、ここでNは1より大きな整数であり、

前記N人の使用者の各々にレート「 R_i 」を割り当てるステップと、ここで、前記レートはi番目の使用者に対して最小限必要なアクセスのレートを示しており、 $1 \leq i \leq N$ 、

i番目の使用者の前記共有リソースにアクセスする資格のしるし「 B_i 」を前記N人の使用者の各々に対して蓄積するステップと、

前記N個の使用者の各々に割り当てられたあらかじめ決められた限度値 L_i まで、その関係付けられたレート R_i で各 B_i を増加するステップと、

他のどの B_i もその関係付けられた限度値 L_i にあって、かつ前記選択された B_i がその限度値 L_i にない場合は、その関係付けられた R_i よりも大きなレートで、選択された B_i を増加させるステップと、そして、

その関係付けられた B_i がゼロでない場合は、その使用者に前記共有リソースへアクセスすることを許可するステップと、

から構成されることを特徴とする方法。

【請求項13】 請求項12記載の方法においてさら

に、前記リソースの*i*番目の使用者にアクセスを許可する際に、*B_i*を削減するステップから構成されることを特徴とする方法。

【請求項14】 請求項12記載の方法において、前記使用者が異なる優先権を割り当てられて、前記選択された*B_i*が前記割り当てられた優先権に基づいて選択されることを特徴とする方法。

【請求項15】 請求項12記載の方法において、前記使用者の各々に関係付けられた限度値*L_i*が同じ値を備えていることを特徴とする方法。

【請求項16】 請求項12記載の方法において、前記使用者に関係付けられた少なくとも2つの限度値*L_i*が異なる値であることを特徴とする方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、制限されたリソースを多数の使用者および/または使用者クラスの間で共用するためのシステムおよび方法に関する。

【0002】

【従来の技術】リソースは多数の使用者によって共用されることが多い。たとえば、電気通信サービスの供給者は、顧客の要求にサービスするために重要なリソースをある限定された数だけしか備えていない。使用されていないリソースを充分に利用させる一方で重要な使用者にシステムの性能低下を起こさせないようにするために、リソースを割り当てるための公平で効率的なシステムが望まれている。

【0003】共用されたリソース環境の中でリソースの公平な共用を保証するという議論は現在のビジネス環境に広がっている。共用は原理的に規模の利益に到達することができること、また需要の非同時性（たとえば、ビジネス地域対住宅地域のトラフィック）が活用できることが広く認識されている。他の顧客からの予想もしない大きな需要に直面しても使用者の契約したリソースの「持ち分」に対してリアルタイムのアクセスを保証するような、完全なフレーム・ワークは存在しない。リソースの完全な分割によって保護を達成することができる。そのような分割は、使用者がお互いから保護されている場合には効果的だが、リソースが特定の使用者に割り当てられていてしかも使用されていない場合には無駄になるので効率的ではない。

【0004】

【発明が解決しようとする課題】典型的なリソースの共用状況においては、通信およびコンピュータのネットワークによって例証されるように、三つの大きな課題について言及する必要がある。すなわち、システムまたはサブシステムへのアクセスを制御する課題、特定のプロセッサを予定に入れる課題、および個別のリソースを割り当てる課題である。

【0005】課題と解決策を視覚化できるようにするた

めに、AT&T社の5ESS[®]交換機を参照する。図1は、ストリームaからストリームzで表される複数のクラスのトラフィックの要求に従う5ESS[®]を図示する。典型的に、ストリームは特定のサービスの要求を束ねており、たとえば呼び出し設定を含むことができる。たとえば、無線サービスに関連したストリームは、ハンドオーバー要求、位置アップデート、その他を含むことができる。一定の活動が、リンク、プロセッサ、データベース、トランクなどのような多数のリソースに要求を配置する。

【0006】5ESS[®]は、ある種の持続されたトラフィック混合に従うあるグレードのサービスを提供することができるように設計され設備されている。基本的なアクセス制御機構は、どのクラスもその契約の値を越えないことを保証するように実行されることになっている。そのようなアプローチは効果的ではあるが、効率的とは言いがたい。結局、他のクラスがその契約のレート以下にあるのでシステムが容量以下で動作しているとき、その割り当てられたレートを越えるクラスへのアクセスは拒絶されてはならない。トラフィックは可変であるのでこれは普通に発生する。一つのクラスがその契約のレートを越えることはありえるが、すべてのクラスがその最大限の持ち分を要求しているときにはこれは必ずしも発生しない。

【0007】第二の課題、すなわちプロセッサの共用は本来ローカルなことである。包括的なアクセス制御に加えてプロセッサの共用を制御することが必要なことがある。たとえば、特定のプロセッサが包括的なリソースとして重要な役割を演ずることがある。第二に、異なるクラスはいろいろなリソースに種々の負担を課す。まさしくその性質によって、包括的なアクセス機構は、あらゆるリソースの各々のクラスの要求の評価に基づいていなければならない。さらに、単一のクラスそれ自体が多数の活動を含むことがある。したがって、この評価は活動のある種の混合を仮定する。しかし、いったんあるクラスがアクセスを得たら、それが将来もそのような混合を持つという保証はない。

【0008】最後に、個別的なリソースの共用の課題に直面する。一例は、5ESS[®]の中のトランクに対する競合である。

【0009】アメリカ特許第5,274,644号（「644特許」：参考のためここで組み入れた）は、共通リソースへのアクセスを制御するための方法のいくつかの事例を開示しており、共通リソースへのアクセスを制御するために「トークン」を利用する機構を含んでいる。「644特許」のシステムは、リソースへの使用者のアクセスがトークンへの使用者のアクセスに基づくように統制する、各使用者に関係付けられた「スロットル」を含んでおり、これを以下に説明する。

【0010】図2は、単一ストリームに対するレート制

御スロットルを図示する。本システムは、トラフィックの重要なパラメーター（到着率、ピーク）がある限度内に保たれるかぎり、望ましいグレードのサービスを提供することができる。レート制御スロットルは、長い目で見れば、システムへの入場許可レートがあらかじめ決められた値を越えないことを保証して、そのピークを制限することによってトラフィックを整えている。このあらかじめ決められた値が、各使用者に対する契約の入場許可レート「 r_i 」（ $i=1\sim N$ 、 N は使用者の合計数である）、およびピークに関連した因子「 L_i 」を確立するために使用される。この機構は、各使用者に対する、有限の容量「 L_i 」のバンク B_i と、あらかじめ決められたレート r_i でトークンを生成する発信源（図示せず）とから構成される。入場許可機構は、次のように動作する。

- 1) トークンは、レート r_i で生成される。
- 2) すでにバンクの中にあるトークンの数が L_i より少ない場合は、新たに生成されたトークンはバンク B_i の中に挿入される。そうでない場合は、トークンは破壊される。

3) バンク B_i の中にトークンを発見するユーザー i からの到着は、サービスに入るのを許可される（そして、そのトークンは破壊される）。そうでない場合は、その到着は拒絶されるかまたは待ち行列に入れられる。

【0011】上に示したように、レート制御スロットルは、各クラスに対して別々のバンクを提供することにより N 個のクラスの使用事例に一般化することができる効果的な機構である。このシステムは効果的ではあるが、完全に効率的ではない。いくつかのクラスが一定のリソースを求めて競合するとき、その内のいくつかがその契約レベル以下で動作することがありそうである。バンクが満杯であることを見つけてトークンが破壊されるレートが、リソースの利用不足の指標である。（もちろんリソースの提供とトークン・レートの割り当てとは正常であると仮定する。）

【0012】利用不足という課題を解決するための一つの提案は、「644特許」の中で記述されているように、予備バンクまたはあふれバンクと名づけられる余分なバンク「 B_0 」を付加することである。通常のバンクと同様に、あふれバンクもまた限定された容量を備えている。本機構は次のように動作する。（図3を参照のこと）。

- 1) クラス i に対応するトークンが契約のレート r_i で生成される。
- 2) すでに B_i の中にあるトークンの数が L_i より少ない場合は、新たに生成されたクラス i のトークンはバンク B_i の中に挿入される。そうでない場合は、トークンはあふれバンク B_0 の中に蓄積される。あふれバンクがたまたま満杯になった場合には、そのトークンは破壊される。

3) バンク B_i の中にトークンを発見するクラス i の到着はサービスに入るのを許可される（そして、そのトークンは破壊される）。バンク B_i の中で利用できるトークンはないが、あふれバンクが空でない場合には、クラス i 到着もまたサービスに入るのを許可される（そしてトークンがあふれバンクの中で破壊される）。そうでない場合は、クラス i の到着は拒絶されるかまたは待ち行列に入れられる。

【0013】BergerとWhittによって著述された「複数クラス入力規制スロットル」という題名の記事（Proceedings of The 29th IEEE Conference On Decision and Control, 1990, pp.2106-2111）で、許可されない到着がシステムから消失される事例を調査している。この記事は、二つのポアソン分布ストリームの事例に対する正確な閉塞とスループットと、 $N>2$ のクラスに対する近似モデルを論議している。（この記事はここでは参照により組み込まれている。）

【0014】すぐには許可されない到着は待ち行列に入れられ、かつクラスの数 2 を越える場合は、待ち行列に入れられたジョブによるあふれバンクへのアクセス規則が規定される必要がある。

【0015】さらに、もう一つの効率的な複数クラスの入場許可制御アルゴリズムは、レート制御スロットルに類似の、リーク・バケツ機構であり、やはり「644特許」の中に記述されている。固定されたレートで流れ込むある量の素材（たとえば、トークン）の利用可能性に注目する代わりに、リーク・バケツ機構は、固定されたレートで排出するバケツの中のスペースの利用可能性を考慮する。

【0016】 N 個の実在のバッファに加えて（待ち行列 Q_1, Q_2, \dots, Q_N ）、リーク・バケツのアクセス機構は、 $N+1$ 個の仮想バッファ（リーク・バケツ $L_{B_0}, L_{B_1}, \dots, L_{B_N}$ ）から構成される。各々のクラスは、それ自身の待ち行列と、それ自身のリーク・バケツとを割り当てられる。待ち行列の大きさは無限である一方で、リーク・バケツは有限の容量である。 L_{B_i} の容量を L_i として示す。ここで、 $i=0, \dots, N$ 。追加のリーク・バケツ（ L_{B_0} ）は、すべてのクラスに共通である。システムに入るのを許可されるために、クラス i の到着は b_i 量の流体をそれ自身の L_{B_i} の中に、または、前者が最大量を含むことができない場合には共通の L_{B_0} に、挿入しなければならない。 L_{B_i} と L_{B_0} の中の予備の容量の和が b_i より少ない場合は、到着は Q_i の中で待つ。クラス i のトラフィックによって起動された流体は、バケツ L_{B_i} の中に、またおそらく L_{B_0} の中に連続的に挿入されて、 Q_i が空（から）でないかぎりスペースがリークによって作られる。ライン到着のヘッドのための必要な量が挿入されるとすぐに、一つのクラス内のアクセスは F I F O 規則（すなわち、先入れ先出し方式）に従って許可される。

LB_i 、 $i \neq 0$ は契約のレート r_i でリークする。バケツ、たとえば LB_j 、 $j \neq 0$ が空のとき、レート r_j でのその放出能力は、直ちに LB_0 へ伝達される。流体が LB_j の中に挿入されるとすぐに、リークはそれに返却される。すなわち、 LB_0 はそれ自身のリークを持たない。

【0017】

【課題を解決するための手段】先行技術の不利益な点は、改善されたリソース配分のシステムおよび方法の特徴とする本発明により解決された。

【0018】一面では、本発明はリソースへのアクセスを要求する複数の使用者の間で共用され限定されたリソースにアクセスを割り当てる方法の特徴としており、この割り当ては、各使用者に関係付けられた最小限の保証されたレートでのリソースへのアクセスを各使用者が許可されるように成される。そして、リソースが利用できる場合だけ、その最小限の保証されたレートを越えている使用者によってリソースが利用できるようにしている。この方法は以下のステップから構成される。各使用者に関係付けられた最小限の保証されたレートによって決定されたレートで各使用者にトークンを割り当て、各使用者に関係付けられたバンクの中に各使用者に割り当てたトークンを蓄積して、各バンクは各使用者に関係付けられた有限の容量を備えていて、ある使用者の関

係付けられたバンクがその容量に到達した場合のその使用者に割り当てたトークンを、別の使用者のバンクがその容量以下にある場合のその別の使用者に割り当てられたバンクの中に蓄積して、そして、その使用者がその関係付けられたバンクの中にあらかじめ決められた数のトークンを備えている場合には、リソースへのアクセスを要求している使用者にリソースへのアクセスを許可する。そうでない場合には、そのリソースへのアクセスを拒絶する。

【0019】好ましい実施例の中では、各使用者は異なる優先権を割り当てられて、その割り当てられた優先権に基づいて選択される。

【0020】この方法はさらに、リソースへのアクセスを許可された使用者に関係付けられたバンクから、あらかじめ決められた数のトークンを削除するステップを含む。あらかじめ決められた数のトークンは、一つのトークンであってもよい。各バンクは、あらかじめ決められた同じ有限の容量を備えているか、または少なくとも二つのバンクが別の容量を備えているかのいずれかである。

【0021】もう一面では、本発明は共用されたリソースへのアクセスを割り当てる方法の特徴としており、下記のステップから構成される。

(a) リソースの複数の使用者の各々に対して加算できるカウンタを確立して、この加算できるカウンタは計数を備えていて、

(b) 本計数が各使用者に関係付けられたあらかじめ定

義された最大計数以下である場合には、各使用者に対して確立されたカウンタの計数を各使用者に関係付けられたレートで増加させて、

(c) アクセスを要求している使用者に対して確立されたカウンタの計数がゼロでない場合は、リソースへのアクセスを要求している使用者にアクセスすることを許可する。そして、

(d) ステップ(b)で増加させられる他のカウンタの中の計数がどれもそのあらかじめ定義された最大計数にある場合は、そのカウンタがあらかじめ定義された最大計数になっていない使用者に関係付けられたカウンタの中でその関係付けられたレートより高いレートでその計数を増加させる。

【0022】なお別の面では、本発明は、N個の使用者の間で共通のリソースへのアクセスを共用するための方法の特徴としており、ここでNは1より大きな整数であり、以下のステップから構成される。

【0023】N個の使用者の各々にレート「 R_i 」を割り当て、このレートはi番目の使用者に対する最小限の要求されるアクセスのレートを示しており、ここで、 $1 < i < N$ である。N個の使用者の各々に対して共通のリソースへアクセスするためにi番目の使用者の資格のしるし「 B_i 」を蓄積する。N個の使用者の各々に割り当てたあらかじめ決められた限度値 L_i まで各 B_i をその関係付けられたレート R_i で増加させる。他の B_i がどれもその関係付けられた限度値 L_i にあり、かつ選択された B_i がその限度値 L_i にない場合に、その選択された B_i をその関係付けられた R_i より大きなレートで増加させ、そして、その関係付けられた B_i がゼロでない場合に、そのi番目の使用者に共通リソースへのアクセスを許可する。望ましくはこの方法はさらに、そのリソースのi番目の使用者にアクセスを許可する際に、 B_i を削減するステップを含む。

【0024】

【発明の実施の形態】本発明の第一の実施例は、上で論議したトークンとリーク・バケツの機構の変形を特徴とする。ここでは、あるクラスは、最初にトークンを共通のバンク B_0 から引き出すことができて(あるいは、 LB_0 にその流体の特定量を挿入することを許可されていて)、そして、その共通バンク(またはバケツ)が満杯であるときだけそれ自身のバンク(またはバケツ)を使用する。この機構は、システムがあるクラスに優先権を与えることを許容して、最初に使用することのできるいくつかのあふれリソースがある場合は、そのクラスが将来の使用のためにリソースへのそれ自身のアクセスを「節約」させることを許容する。

【0025】第二の実施例では、システムはあふれバンクやバケツを完全に省略することができて、その代わりに、ある種の選択された使用者に対して、その契約した容量以下で動作している使用者によって生成された超過

のトークン（またはリーク容量）を分配し直すことができる。この機構は以下に説明するが、説明を簡単にするために、リーク・バケツに類似の状況の中だけで説明される。

【0026】図4に示した機構を考察すると、共通バケツLB₀の大きさは任意に小さく作成することができ、このことは許可されたストリームのバースト（突発）時の共通バケツのインパクトが無視できることを意味している。図4は、バケツLB₂が空という特定の状況を捕らえている。したがって、その排出能力は現在、共通バケツLB₀で行使されている。また、実在の待ち行列Q_iの中で待っているクラス1の到着がある点に注意のこと。図4に示すように、クラス1の到着に対してシステムへのアクセスを許可するために要求された特定量を挿入するためのバケツLB₀とLB₁とに余地がないことをこれは意味する。（このように、クラス1の到*

*着はQ_iで待っている。）

【0027】補助変数Z_n⁽ⁱ⁾, i=1, 2で、その到着をカウントしない到着時刻nでの、LB₀、LB_i、Q_iの中の作業負荷の総和を示すこととする。また、その待ち行列とリーク・バケツの中の作業量として、Q_i、LB_i、LB₀を示す。図4に示した特定の状態の中で、Z_i=LB₀+LB_i+Q_i、Z_n⁽²⁾=LB₀。流体の特定量b_iの挿入が完了するまで到着はQ_iの中で待たなくてはならない。しかし流体は、Q_i>0、LB_i+LB₀-L_i-L₀<0であるかぎり、バケツからリークされる。また、LB_i>0=>LB₀=L₀に注意のこと。システムへの第n番目の到着の時刻を示すのに索引nを使用し、時刻nと時刻n+1の間の時間をΔ_{n+1}で示し、確率変数を導入する。

【数1】

$$a_n^{(1)} = \begin{cases} 1 & \text{時刻 } n \text{ での到着がクラス } 1 \text{ である場合} \\ 0 & \text{その他の場合} \end{cases}$$

【0028】同一時刻の到着は許されない。すなわち、20※含める。それから間隔Δ_{n+1}の間の排出を計算する。
a_n⁽¹⁾ a_n⁽²⁾ = 0。到着時刻の間のシステムの展開

【数2】

を記述するために、前の到着により寄与した作業負荷を※

$$\hat{Z}_{n+1}^{(1)} = Z_n^{(1)} + a_n^{(1)} b_n^{(1)} + \min(a_n^{(2)} b_n^{(2)}, [L_0 - Z_n^{(2)}]^+) \quad (1)$$

$$\hat{Z}_{n+1}^{(2)} = Z_n^{(2)} + a_n^{(2)} b_n^{(2)} + \min(a_n^{(1)} b_n^{(1)}, [L_0 - Z_n^{(1)}]^+) \quad (2)$$

$$Z_{n+1}^{(1)} = [\hat{Z}_{n+1}^{(1)} - \Delta_{n+1} r^{(1)} - [\Delta_{n+1} r^{(2)} - \hat{Z}_{n+1}^{(2)} + L_0]^+]^+ \quad (3)$$

$$Z_{n+1}^{(2)} = [\hat{Z}_{n+1}^{(2)} - \Delta_{n+1} r^{(2)} - [\Delta_{n+1} r^{(1)} - \hat{Z}_{n+1}^{(1)} + L_0]^+]^+ \quad (4)$$

【0029】方程式1-4は共通バケツLB₀を持つことのインパクトを強調する。特定のバケツ、たとえばLB₁の展開を検討する。方程式3の中のマイナスの項はLB₂による共通の排出への寄与を表す。Δ_{n+1} r⁽²⁾、つまりその間隔の間のその排水能力が、それ自身の排他的な作業負荷Z_{n+1}⁽²⁾ - L₀を越えるとき、後者は共通の排出に寄与する。この共通の排出能力が、L₀を持つことのきわだった特徴である。一方、方程式3の右★

$$Pr\{Z_{n+1}^{(1)} > x\} = Pr\{Z_n^{(1)} > L_0 + L_1 + x\}.$$

(5)

方程式1と2とは結合される。すなわち、Z_n⁽ⁱ⁾, i=1, 2のCDFを得ることは、(Z_n⁽¹⁾ Z_n⁽²⁾)の結合分布を見つけることを必要とする。状態空間が、3というよりはむしろ2であるということに注意のこと、我々が共通のバケツと自分自身のバケツに挿入する際に順序を逆にする場合にそうなる。このことは、D_n⁽ⁱ⁾, i=1, 2のCDFを計算するための数式手順の中で相当な簡略化を表す。

【0030】それを共通バケツなしの複数クラスのアクセスと比較することによって、アクセス方式に対するか

30★辺の第三項によって示されるように、Z_{n+1}⁽¹⁾はクラス2の到着によっていくぶん影響を与えられる。しかし、この項が示すように、インパクトはL₀を越えることができないために、我々の関心は、到着時刻nでのQ_i, i=1, 2の中での作業負荷D_n⁽ⁱ⁾の累積分散関数(CDF)を見つけることである。それゆえに、

【数3】

40 なるの洞察を得ることができる。上記したように、そのような機構は保護の観点から効果的だが、効率が欠けている。ここに、二つのクラスという特定の事例に対するポイントを証明する。この目的のために、我々のオリジナルのシステムと同じ到着に関わる共通バケツのない別のアクセス機構を検討して、その変数に差異を付けるためにバーを使用する。特に、方程式1と2とは、次式に縮小できて、

【数4】

$$Z_{n+1}^{(i)} = [Z_n^{(i)} + a_n^{(i)} b_n^{(i)} - \Delta_{n+1} r^{(i)}]^{+}, i=1, 2. \quad (6)$$

そして、方程式5を下記に変換する。

$$Pr(D_n^{(i)} > x) = Pr(Z_n^{(i)} > L_i + x). \quad (7)$$

【0031】

【外1】

$L_i = L_1, i=1, 2, L_0 = \epsilon$ を選び、そして、共通バケツの大きさへの作

業負荷の依存を強調するために、 $Z_n^{(i)}(\epsilon)$ を書く。

$$W_n^{(i)} = \lim_{\epsilon \rightarrow 0} Z_n^{(i)}(\epsilon) \quad \text{として、} \quad W_0^{(i)} = Z_0^{(i)}, \quad i=1, 2 \text{ とおく。}$$

すると、任意の整数 $n > 0$ に対して、確率変数 $W_n^{(i)}$ は $Z_n^{(i)}$ より確立論的に小

さい。すなわち、すべての $X \geq 0$ に対して、 $Pr(W_n^{(i)} > x) \leq Pr(Z_n^{(i)} > x)$

【0032】 $L_0 \rightarrow 0$ になると、クラス i (方程式3と *ラスは、 $Z_n^{(i)}$ の排出に寄与する (方程式3と4)。
4) の作業負荷への他のクラスの寄与がゼロに減少する 【外2】

点に注意のこと。しかし、 $L_0 = 0$ に対しても、他のク*

したがって、限度値の中で、 $Z_n^{(i)}(\epsilon)$ の展開は、正でない項の加算で $Z_n^{(i)}$

の展開と一致する。

【0033】 以下の推論は、共通バケツを備える複数ク ※示す。

ラスのアクセス制御装置はアクセス遅延という見地から

【0034】

古典的なアクセス制御装置より性能が優れていることを※ 【外3】

上述した条件のもとで、確率変数 $D_n^{(i)}$ は、すべての $X \geq 0$ に対して $D_n^{(i)}$

より小さい。すなわち、 $Pr(D_n^{(i)} > x) \leq Pr(D_n^{(i)} > x)$

この不等式は再度書き直すことができ、

【数6】

$$Pr(W_n^{(i)} > L_i + x) \leq Pr(Z_n^{(i)} > L_i + x)$$

これは上記の提案に従ったままである。

【0035】 上記の注記のように、使用者が余分なバケツ LB_0 を最初に調査して、 LB_0 が満杯である場合にはそれ自身のバケツの中に流体を挿入するという、古典的なリーク・バケツ機構への変更が提案された。それから、提案された機構の動作は標準の多重アクセスのアプローチと比較された。一貫した比較を行うために、バケツ LB_0 の大きさはゼロに設定された。そうすることのプロセスの中で、余分な容量を分配し直すための機構は、主としてリーク能力の再配布に依存しており、余分なバケツ LB_0 には依存しないことが明確になった。それゆえに、余分なバケツは全く省略できて、アルゴリズムは以下のように修正される。バケツ LB_i が空であるときはいつでも、レート r_i でのその排出するための能力は、空でないバケツの間で分配される。リークを再分配するために二つのアルゴリズムを使用することができる。

30 1) 契約のレートに比例した超過容量の再分配。 LB_i が空だとする。 $A = \{LB_i : LB_i > 0\}$ とおく。次に、バケツ LB_i のリークのレート $i \in A$ を

【数7】

$$r_i = \frac{r_i}{\sum_{j \in A} r_j}.$$

だけ増加させる。

2) 優先権テーブルに従った超過容量の再分配。削減する優先権順にバケツを順序づける。使用されていないリークは、最初に優先度リストの先頭になる。このバケツが空であるときだけ、それはリストの第二番目へ移動する。以下同様である。

【0036】 この二つの機構が、公正さの特定の解釈から差別的な配置へと広範囲にわたるレンジの両極端である。他の可能性は、最も要求の高いバケツに超過分を割り当てるが含まれる。

【0037】 本発明のこの実施例に対して少くとも二つの注目すべき効果がある。第一に、レートはシステムへの平均到着率に関係するが、バケツの大きさはシステムが耐えることのできるバースト (突発性) に関係付けら

れる。その最悪バーストが、標準のリーク・バケツをクラスあたりに割り当てる場合のバーストに等しいという機構を備えていることは有利である。第二に、超過容量の分配に焦点をしばった、より巧みな実現が達成される。

【0038】システムの故障、あるいはスロットルによって制御されない外部トラフィックは、なお時折システム過負荷のシナリオに陥ることがある。レート制御スロットルは閉ループ制御装置である一方で、レート $\{r_i\}$ のセットをシステムの状態に従属させることによって、遅い時間・規模の順応ループを提供することが可能となる。

【0039】上に提案された余分なバケツのないアクセス制御装置は、余分な容量を分配するためにレートの組み込み式上方適応機構を備えていて、閉じたループを持つ過負荷時の遅い時間・規模の下方適応とは対照的である。実際に、制御されてないトラフィックに起因する混雑のために全体のレートが減少される一方で、不活動な使用者の寄与のためにいくつかのクラスに関係付けられた現実のレートが増加されることは、確かに可能である。

【0040】以前のアクセス制御機構は、一群のリソースへの複数のレベルでのアクセスを制御するために使用することができる。たとえば、一つのシステムを使用して多数の顧客の間のサービス（たとえば無線）へのアクセスを制御して、それからもう一つの機構を使用してリソースに対する多数の異なるサービスの間でアクセスを制御するのは望ましいことである。図5はそのようなシステムを示しており、ここでは多数のセグメント $a-z$ がサービス $1-N$ の中の一つへのアクセスのために優先権セグメント・アクセス制御装置 (PSAC) 20 の制御の下に最初に競合する。その後、複数サービス・アクセス制御装置 (MSAC) 30 の制御の下に、サービスはリソース 40 へのアクセスを争う。MSAC 30 は、望ましくは上で議論されたアクセス制御機構の一つを使用する。（便宜上、「トークン」類似を使用して記述したが、リーク・バケツ類似もまた使用してもよい）。

【0041】各々の PSAC 20 は以下の好ましい方法に従ってその対応するサービスへのアクセスを制御する。図6を参照すると、多数のバンク $B_1 - B_N$ が示されており、各々はシステムの利用者の N クラスの一つに対応する（各クラスは利用者セグメント $a-z$ に対応している）。各クラスのバンク B_i はトークンのストリームを契約のレート r_i で受け取る。また、あふれバンク O_i は各クラスと関係付けられている。特定のクラスに対して生成されたトークンが満杯のバンク B_i を発見した場合、そのトークンはそのクラスのあふれバンク O_i に直ちには挿入されない。その代わりに、システムは最初に余分なトークンをバンク O_i に挿入しようとする。バンク O_i はクラス i のあふれバンクを示し、このシ

テムの中のより高い優先順位クラスである。あふれバンク O_i が満杯である場合には、該トークンはあふれバンク O_2 に挿入される。以下同様に続く。

【0042】このように、各クラスがその契約のレートを保証される一方で、超過容量はあるクラスに味方する優先権機構に分配される。

【0043】要求された優先権順位機構が実行されるのに適切のように、この機構を変更することができる。たとえば、単一の高優先順位クラスは、あふれトークンをこのクラスのあふれバンクに最初に挿入しようとしている各々のクラスに指名することができる、そしてこのあふれバンクが満杯の場合には、それ自身のあふれバンクの中に超過トークンを挿入する。あふれバンクの大きさはまた、システムの好ましい優先権を反映するために変更することができる。

【0044】プロセッサ共用のための複数クラスのオーリング・アルゴリズムをこれから記述する。 N 個の利用者によって共用されるプロセッサについて考察する。プロセッサのリアルタイムの部分 $f^{(0)}$ がオーバーヘッドに割り当てられるとする。（典型的に $f^{(0)} \approx 0.1 \sim 0.15$ ）オーバーヘッドは、総計 $N+1$ 個の利用者に対して 使用者 0 と見なされる。

【0045】使用者 i によって契約されたリアルタイム使用の部分 $f^{(i)}$ を示すとする。 $\sum_{i=0}^N f^{(i)} = 1$ と仮定する。契約した部分と要求されたオーバーヘッド $f^{(0)}$ を加えた総和が 1 より少ない事例については、正規化に注意が必要である。使用者 i のジョブ要求は、バッファ B_i の中に蓄積される。バッファ B_0 は、オーバーヘッド作業のために予約される。時間は、持続時間 τ のスロットに分割される。細分性はプロセッサに依存しており、そのトレードオフは識別するのが簡単である。細分性が粗すぎるのは制御がしにくくなり、細分性を微細にしすぎると容認できないオーバーヘッドをもたらすことがある。

【0046】 γ 秒ごとにプロセッサが、バッファ B_i のセット $B = \{B_i, 0 \leq i \leq N\}$ の一つから γ 秒に値する作業を拾い上げなければならない。プロセッサは下記のような方法で予定されている。

a) その契約の限度内で動作するあらゆる使用者はそれが期待するサービスを受け取ることを保証する。

b) その契約の限度を越えて動作するあらゆる使用者は少なくともその契約上の持分を受け取ることを保証する。そして、

c) 作業がいずれかのバッファにある場合は、プロセッサはどの巡回の中でもサービスを拒否しない。サービス規律は、プロセッサが、 γ 秒以下しか値しない作業でバッファをサービスする点において本当は作業を保護しないことに注意のこと。また、 $\gamma \rightarrow 0$ にすると、サービス規律は作業を保護するようになる点にも注意のこ

15

16

【0047】時刻 \underline{n} で、制御変数 $U_n^{(i)}$ は、バッファ B_i に関する決定を示す。 * 【数8】

$$u_n^{(i)} = \begin{cases} 1 & B_i \text{ が選択された場合} \\ 0 & \text{そうでない場合} \end{cases} \quad (8)$$

どのバッファ (B_0 を含む) も空にならないシナリオを考慮する。 $\sum_{j=1}^n U_j^{(i)}$ は、プロセッサが、時刻 \underline{n} を含めて \underline{n} まで、バッファ B_i からセグメントを選ん※ $f^{(i)}$ に適合するために、下記を要求する。 【数9】

$$\lim_{n \rightarrow \infty} \frac{\sum_{j=1}^n u_j^{(i)}}{n} \geq f^{(i)}, i=0, 1, \dots, N. \quad (9)$$

【0048】この長期にわたる平均化手段は必要だが、使用者の期待に適合するには十分ではない。それから、下記によって測定されるような目標値から使用者 i への★ $e_n^{(i)}$ $f^{(i)}$ 実際の配置時刻の最大偏差を最小にしようとする。 【数10】

$$e_n^{(i)} = \sum_{j=1}^n [f^{(i)} - u_j^{(i)}], \quad (10)$$

これは帰納的にアップデートすることができて、 【数11】

$$e_0^{(i)} = 0, \quad (11)$$

$$e_{n+1}^{(i)} = e_n^{(i)} + [f^{(i)} - u_{n+1}^{(i)}], n=0, 1, \dots \quad (12)$$

【0049】時刻 n でサービスされたバッファを b_n * ☆1) 初期状態設定：
で示して、以下のアルゴリズムを検討すると、 ☆ 【数12】

$$e_0^{(i)} = 0 \quad i=0, 1, \dots, N, \quad n=0. \quad (13)$$

2) ポーリング時

【外4】

a) ポーリング決定：

$$b_n = \operatorname{argmax}_i e_{n-1}^{(i)} \quad (14)$$

(辞書式順序との連結を切る)

$$u_n^{(i)} = \begin{cases} 1 & b_n = i \text{ の場合} \\ 0 & \text{そうでない場合} \end{cases}$$

b) 状態アップデート：

$$e_n^{(i)} = e_{n-1}^{(i)} + [f^{(i)} - u_n^{(i)}]. \quad (15)$$

【0050】したがって、好ましい実施例の中で、時刻 \underline{n} でのポーリング決定は時刻 $\underline{n}-1$ で最も大きな $e^{(i)}$ を選択することによって成される。このように、プロセッサへのアクセスは、その契約した持ち分とその実際の使用オーバー時間の間の最も大きな差を示す使用者に許可される。これは、すべての使用者がプロセッサに連続的にアクセスを要求しているとき（すなわち、使用者のバッファが決して空にならない時）の、プロセッサ共有の最も公正な方法である。

【0051】しかし、最も現実的なシナリオでは、サーバの利用は1以下でありバッファが空になる。また、ジョブは通常、セグメントに分割される。セグメントの長

さはCPUがタスクを実行するために取る時間を示し、したがって、すべてのセグメントが同じ持続時間を持つと仮定することは非現実的である。次に、これらの現実的な課題を考慮した別の実施例を検討する。

【0052】 $S_n^{(i)}$ をポーリング時刻 \underline{n} でのバッファ B_i のヘッドでのセグメントの長さとして、そのセグメントの長さに対してある種の上限 S_{\max} があると仮定する。二つの特定の事例、すなわち異質の過負荷と同質の通常の負荷のシナリオに焦点をあてるのが実証となる。過負荷で異質の事例は、方程式15の単純な変更によって、簡単に扱うことができる。

【数13】

$$e_n^{(i)} = e_{n-1}^{(i)} + S_n^{(b_n)} [f^{(i)} - u_n^{(i)}]$$

【0053】すなわち、変更は、ポーリングされたバッファの先頭で、セグメントをサービスしているプロセッサによって費やされる時間によって、クレジット（貸方） $f^{(i)}$ と装置借方 $u_n^{(i)}$ の両者を重み付けすることである。

【0054】同質の通常負荷の事例を考慮すると、各々のクラス i がCPUの同じ部分、 $f^{(i)} = 1/N+1$ 、 $i = 0, 1, \dots, N$ を割り当てられると仮定する。セグメントの長さの分布は、すべてのクラスに対して同じである。このシナリオの中で、自然で、公正で、かつ最も合理的な基準に従う本当に最適な方針は、ラウンド・ロビン・アルゴリズムであり、そこではサーバは空のバッファをスキップする。

【0055】ポーリング・バッファが空であるとき、システムは下記のように処理する。プロセッサがセグメントの到着を待つことは無駄であるので、空のバッファはスキップされると決定する。それから、問題は空のバッファへの「仮想」の巡回をどのように説明するかということである。仮想巡回に対して空のバッファを満たさないことを考慮してもよく、それに経過時間に対してクレジットを与え続けてもよい。このアプローチの持つ課*

a) ポーリング決定

$$b_n = \underset{i}{\operatorname{argmax}} e_{n-1}^{(i)}$$

（辞書式順序との連結を切る）

$$u_n^{(i)} = \begin{cases} 1 & b_n = i \text{ の場合} \\ 0 & \text{そうでない場合} \end{cases}$$

（ i は仮想セグメントであり、ここではプロセッサは時間を費やさない。）

b) 状態アップデート

$$e_n^{(i)} = e_{n-1}^{(i)} + S_n^{(b_n)} [f^{(i)} - u_n^{(i)}].$$

【0057】上述したように、たとえリソースへのアクセスが統制されるとしても、それにもかかわらず一つ以上の特定の個別のリソースへのアクセスを統制することは望ましいことである。個別のリソースにアクセスするための複数クラスの巡回トークン機構について次に説明する。

【0058】トランク、3者会議回線、UTD（汎用トーン・デコーダ）は、個別のリソースの中にあり、異なるクラスの利用者の間で共有される必要がある。そのま

*題は、不活動な発信源によって累積されたクレジットが、それが目覚めたとき、CPUを独占することがありうるということである。合理的な解決策は帰納的に指数的な平滑化を導入することであり、それによって過去よりも現在に多くの重み付けがなされる。もう一つの選択肢は、活動的なバッファの中だけで各段階でクレジットを配布することである。別のアプローチは、選択されたバッファが仮想回線に対する「失われた機会」を負担させられるという概念に基づいている。

【0056】今や、一般的なプロセスを述べることができる。 $S_n^{(i)}$ がポーリング時刻 n でのバッファ B_i の先頭でのセグメントの長さであることを思い出そう。バッファ B_i が空であるとする、これを S_i にセットする。ここで、 S_i つまり利用者 i の平均のセグメント長は下記のとおりであり、これを仮想セグメントと呼ぶ、

1) 初期状態設定:

【数14】

$$e_0^{(i)} = 0 \quad i=0, 1, \dots, N, \\ n = 0.$$

2) ポーリング時

【外5】

さしくその本質によって、これらのリソースは、ビジーであるかまたは動作していない。通常、リソースの保持時間は、その状態を伝送するために必要とされる時間をはるかに越える。そしてもちろん、リソースの個別の性質は正の整数に写像される。以下は、共通のプールをもつ巡回トークン機構である。

【0059】 N 個の利用者が競合する M 個の装置のリソースがあるとする。

50 仮定、

1. n_i のトークンをクラス i のバンクに割り当てる。

$i = 1, 2, \dots, N$

2. n_0 のトークンを共通のバンク b_0 に割り当てる。

したがって、

【数15】

$$n_0 + \sum_{i=1}^N n_i = M$$

【0060】各々のトークンがリソースの一つの装置にアクセスを与える。トークンは巡回されるが、破壊され
ない。すなわち、アクセスを許可した装置がフリーにな
ると、取られたバンク（自分自身のバンクまたは共通バ
ンク）にトークンは返却される。

【0061】たとえば、クラス i が $m \geq 1$ 個の装置のリ
ソースを必要すると仮定しよう。3つの可能性がある。

(1) それぞれ自分の m 個の不動作中のトークンを持
っていて、この場合、トークンと引き換えに要求する装
置へのアクセスを得る。(2) それはそれ自身の m 個以
下のトークンしか持っていないが、共通バンクの中のト
ークンは要求を完了するのに十分であり、クラス i が、
(1) のように処理する。そして、(3) それ自身の利用
できるトークンと共通バンクの利用できるトークンと
の和が m 個以下の場合、システム特有の設計によっ
て、その要求が拒絶されたり待ち合わせされたりする。

【0062】この機構は、リソースの完全な分割から全
面的な共用まで、広範囲にわたる解決策をカバーする。
トークンが共通バンク ($n_0 = 0$) に一つも割り当てら
れないとき、完全なリソース分割に変わる。他の極端で
は、 $n_0 = N$ (したがって、 $n_i = 0, i = 0$) の場合、
完全にリソースを共用する。完全な分割は効果的であ
る。クラス i は常に、 n_i 個の装置のリソースへの同時
アクセスを保証される。残っている $N - n_i$ 個のすべ
ての装置が動作していない場合でも、 n_i 個以上のアクセ
スを全く得ることができないことは効率的でない。一
方、完全な共用では、この最後のシナリオを見事に処理
するが、 n_i 個の装置へのあらゆる時刻でのクラス i の
アクセスを保証しない。中間の状況 $0 < n_0 < N$ は、保
証と柔軟性との間でのトレードオフを表す。共通のプ
ールに寄与するために、クラス i は $n_i < n_i$ に配置され
ることになる。同時に使用することのできる装置の最大
数はシステムに集合された負荷によって、 n_i と $n_i +$
 n_0 との間で変化する。

【0063】機構の変更の一つは、トークンをそれ自身
のバンクの中で探索する前に使用者のサブセット（高い
優先順位クラス）に共通バンクへのアクセスを与えるこ
とから成る。正確な応答を得るために、トークンを返却
するための規則の変更が必要とされる。この変更に従う
と、トークンは常に優先権の高いバンクに返却されて、
そこが共通バンクにあふれ出る。これらの変更は、一般
にトークンの巡回方針が下記によって決定されることを

示す。

- トークンとトークン・バンクとを異なるクラスへ割
り当てる。
- 一つまたは二つ以上の共有されたバンクにトークン
を割り当てる。
- バンクへのアクセスを許可する規則。
- 使用されたトークンをどのバンクへ返却しなければ
ならないかを決定する規則。

【0064】複数クラスのレートへのアクセス制御と巡回
トークン機構の間にはいくつかの類似性がある。両方の
アプローチは、トークンの概念、および共通の装置と同
様にクラスに特有の装置の概念という両方の観点から説
明される。しかし、それらが向かう課題の間の差異を明
確にする重要な違いがある。トークンを巡回させる閉ル
ープ特性は、複数クラスのアクセス制御の開ループ特性と
対照をなす。また、前者のトークンは本来的に個別的で
あるが、後者は実数値のトークンを認める。トークン巡
回機構の属性は、個別的なリソースの明快な本質から生
じる。装置は使用中かまたは不動作中であり、しかもこ
の状態は多くの場合リアルタイムで利用できる。一方、
図1で描写されたようなグローバルなシステムに入るの
を許可される到着は、より巧妙な方法で多数の異なるリ
ソースを消費するように連結される。これは、ある種の
プロセッサの利用法を増加させ、ある種の待ち行列等
の中で遅延を増大させる。また、これらの内部リソース
(プロセッサ、リンク、その他)、のインパクトは、
到着と関係付けられた活動の保持時間の間の異なるポイ
ントで発生する。

【0065】以上の説明は、事例としてだけを意図した
好ましい実施例の図を含んでおり、添付の請求項の適用
範囲内で種々の他の実施例が可能である。

【図面の簡単な説明】

【図1】アクセスを求めている複数のクラスを持つAT
&Tの5ESS[®] 交換機のブロック図である。

【図2】先行技術のスロットル制御技法を示す図であ
る。

【図3】第二の先行技術のスロットル制御技法を示すブ
ロック図である。

【図4】第三の先行技術のスロットル制御技法を示すブ
ロック図である。

【図5】本発明のアクセス制御技法を示すブロック図で
ある。

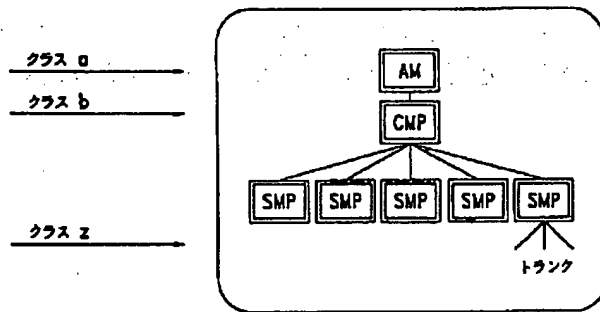
【図6】多数のセグメント間のサービスへのアクセスと
多数のサービスの間のリソースへのアクセスとの両者を
制御するための、本発明に従ったシステムを示すブロッ
ク図である。

【符号の説明】

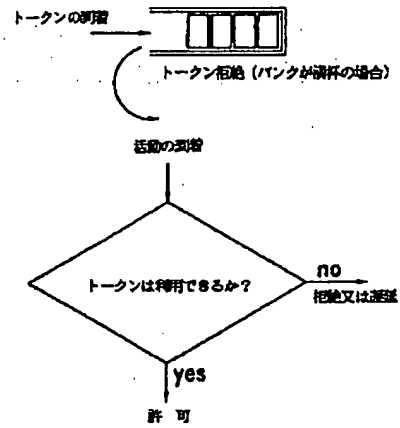
20 PSAC (優先セグメント・アクセス制御装
置)

30 MSAC (複数サービス・アクセス制御装置)

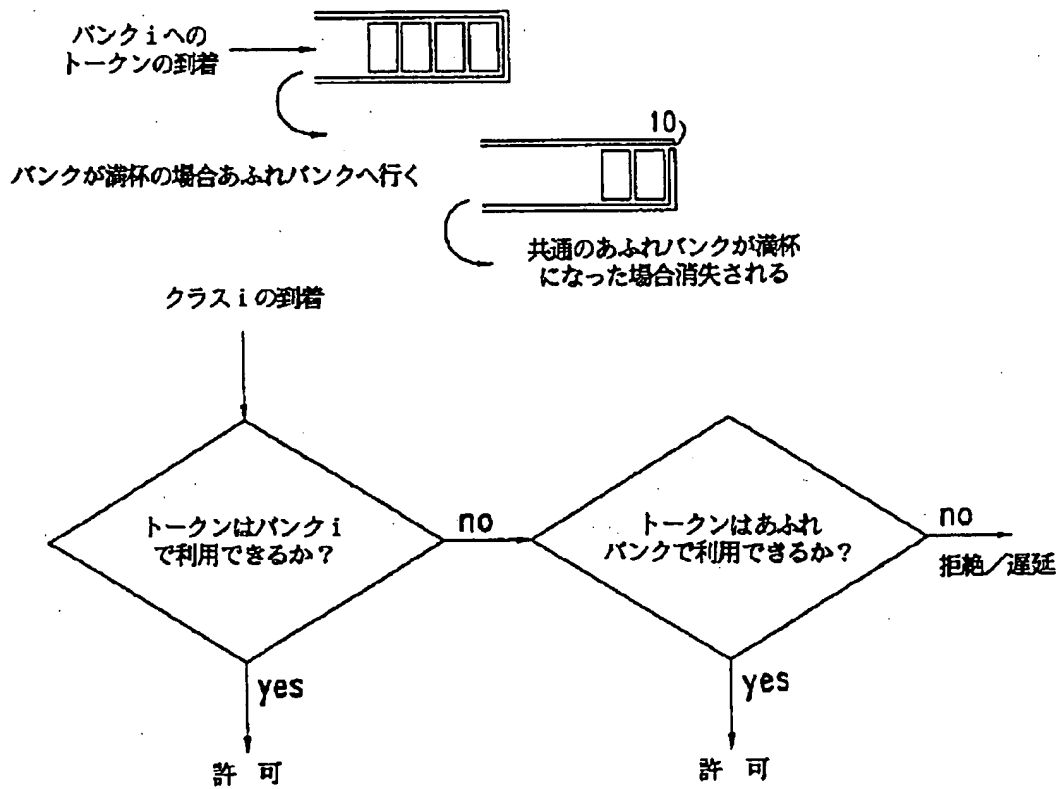
【図1】



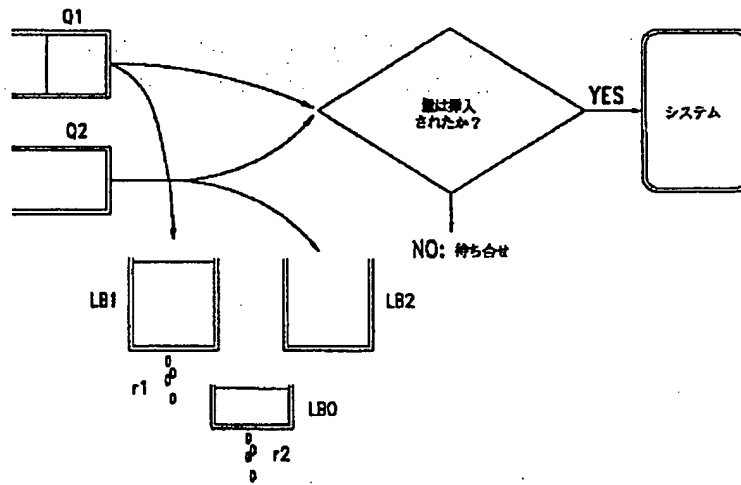
【図2】



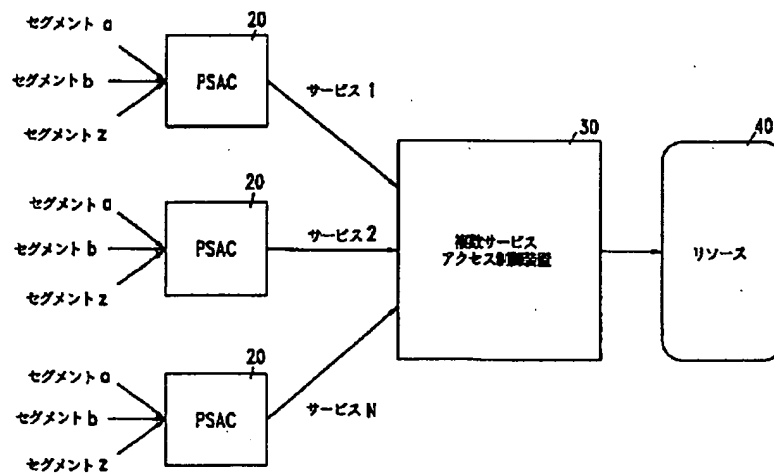
【図3】



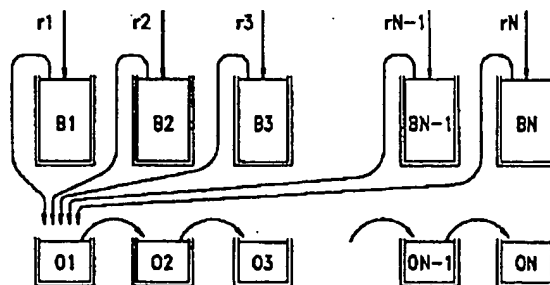
【図4】



【図5】



【図6】



フロントページの続き

(51) Int. Cl.⁶

H04Q 3/545

識別記号

庁内整理番号

9744-5K

9744-5K

FI

H04L 11/20

技術表示箇所

G

102C,